Sparsification of Binary CSPs

Silvia Butti
Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain
silvia.butti@upf.edu

Stanislav Živný
Department of Computer Science, University of Oxford, UK
standa.zivny@cs.ox.ac.uk

Abstract

A cut ε-sparsifier of a weighted graph G is a re-weighted subgraph of G of (quasi)linear size that preserves the size of all cuts up to a multiplicative factor of ε. Since their introduction by Benczúr and Karger [STOC'96], cut sparsifiers have proved extremely influential and found various applications. Going beyond cut sparsifiers, Filtser and Krauthgamer [SIDMA'17] gave a precise classification of which binary Boolean CSPs are sparsifiable. In this paper, we extend their result to binary CSPs on arbitrary finite domains.

2012 ACM Subject Classification Theory of Computation → Graph algorithms analysis

Keywords and phrases constraint satisfaction problems, minimum cuts, sparsification

Digital Object Identifier 10.4230/LIPIcs.STACS.2019.17

Funding This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714532). The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein.
Silvia Butti: Work mostly done while at the University of Oxford.
Stanislav Živný: Stanislav Živný was supported by a Royal Society University Research Fellowship.

1 Introduction

The pioneering work of Benczúr and Karger [4] showed that every edge-weighted undirected graph $G = (V, E, w)$ admits a cut-sparsifier. In particular, assuming that the edge weights are positive, for every $0 < \varepsilon < 1$ there exists (and in fact can be found efficiently) a re-weighted subgraph $G_\varepsilon = (V, E_\varepsilon \subseteq E, w_\varepsilon)$ of G with $|E_\varepsilon| = O(\varepsilon^{-2} n \log n)$ edges such that

$$\forall S \subseteq V, \qquad \mathsf{Cut}_{G_\varepsilon}(S) \in (1 \pm \varepsilon)\mathsf{Cut}_G(S),$$

where $n = |V|$ and $\mathsf{Cut}_G(S)$ denotes the total weight of edges in G with exactly one endpoint in S. The bound on the number of edges was later improved to $O(\varepsilon^{-2} n)$ by Batson, Spielman, and Srivastava [3]. Moreover, the bound $O(\varepsilon^{-2} n)$ is known to be tight by the work of Andoni, Chen, Krauthgamer, Qin, Woodruff, and Zhang [2].

The original motivation for cut sparsification was to speed up algorithms for cut problems and graph problems more generally. The idea turned out to be very influential, with several generalisations and extensions, including, for instance, sketching [1, 2], sparsifiers for cuts in hypergraphs [9, 11], and spectral sparsification [15, 14, 13, 8, 12].

Filtser and Krauthgamer [7] considered the following natural question: which binary Boolean CSPs are sparsifiable? In order to state their results as well as our new results, we will now define binary constraint satisfaction problems.

An instance of the binary[1] *constraint satisfaction problem* (CSP) is a quadruple $I = (V, D, \Pi, w)$, where V is a set of variables, D is a finite set called the *domain*,[2] Π is a set of constraints, and $w : \Pi \to \mathbb{R}_+$ are positive weights for the constraints. Each constraint $\pi \in \Pi$ is a pair $((u, v), P)$, where $(u, v) \in V^2$, called the constraint *scope*, is a pair of distinct variables from V, and $P : D^2 \to \{0, 1\}$ is a binary predicate. A CSP instance is called *Boolean* if $|D| = 2$, i.e., if the domain is of size two.[3]

For a fixed binary predicate P, we denote by $\mathrm{CSP}(P)$ the class of CSP instances in which all constraints use the predicate P. Note that if we take $D = \{0, 1\}$ and P defined on D^2 by $P(x, y) = 1$ iff $x \neq y$ then $\mathrm{CSP}(P)$ corresponds to the cut problem.

We say that a constraint $\pi = ((u, v), P)$ is *satisfied* by an assignment $A : V \to D$ if $P(A(u), A(v)) = 1$. The value of an instance $I = (V, D, \Pi, w)$ under an assignment $A : V \to D$ is defined to be the total weight of satisfied constraints:

$$\mathrm{Val}_I(A) = \sum_{\pi=((u,v),P)\in\Pi} w(\pi)P(A(u), A(v)).$$

For $0 < \varepsilon < 1$, an ε-sparsifier of $I = (V, D, \Pi, w)$ is a re-weighted subinstance $I_\varepsilon = (V, D, \Pi_\varepsilon \subseteq \Pi, w_\varepsilon)$ of I such that

$$\forall A : V \to D, \qquad \mathrm{Val}_{I_\varepsilon}(A) \in (1 \pm \varepsilon)\,\mathrm{Val}_I(A).$$

The goal is to obtain a sparsifier with the minimum number of constraints, i.e., $|\Pi_\varepsilon|$.

A binary predicate P is called *sparsifiable* if for every instance $I \in \mathrm{CSP}(P)$ on $n = |V|$ variables and for every $0 < \varepsilon < 1$ there is an ε-sparsifier for I with $O(\varepsilon^{-2} n)$ constraints.

We call a (not necessarily Boolean or binary) predicate P a *singleton* if $|P^{-1}(1)| = 1$.

Filtser and Krauthgamer showed, among other results, the following classification. Let P be a binary Boolean predicate. Then, P is sparsifiable if and only if P is not a singleton.[4] In other words, the only predicates that are not sparsifiable are those with support of size one.

Contributions. As our main contribution, we identify in Theorem 2 the precise borderline of sparsifiability for binary predicates on arbitrary finite domains, thus extending the work from [7] on Boolean predicates. Let P be a binary predicate defined on an arbitrary finite domain D. Then, P is sparsifiable if and only if P does not "contain" a singleton subpredicate. More precisely, we say that P "contains" a singleton subpredicate if there are two (not necessarily disjoint) subdomains $B, C \subseteq D$ with $|B| = |C| = 2$ such that the restriction of P onto $B \times C$ is a singleton predicate.

The crux of Theorem 2 is the sparsifiability part, which is established by a reduction to cut sparsifiers. Unlike in the classification of binary Boolean predicates from [7], we do not rely on a case analysis that differs for different sparsifiable predicates but instead give a simpler argument for all sparsifiable predicates. The idea is to reduce (the graph of) any CSP instance, as was done in [7], via the so-called bipartite double cover [5]. However, there is no natural assignment in the reduced graph (as it was in the Boolean case in [7]). In order to overcome this, we define a graph G^P whose edges correspond to the support of the

[1] Some papers use the term *two-variable.*
[2] Some papers use the term *alphabet.*
[3] Some papers use the term *binary* to mean domains of size two. In this paper, *Boolean* always refers to a domain of size two and *binary* always refers to the arity of the constraint(s).
[4] Filtser and Krauthgamer use the term *valued CSPs* for what we defined as CSPs. We prefer CSPs in order to distinguish them from the much more general framework of valued CSPs studied in [10].

predicate P. Using a simple combinatorial argument, we show (in Proposition 7) that, under the assumption that P does not "contain" a singleton subpredicate, the bipartite complement of G^P is a collection of bipartite cliques. This special structure allows us to find a good assignment in the reduced graph.

In view of Filtser and Krauthgamer's work [7], one might conjecture that P is sparsifiable if and only if P is not a singleton. While it is easy to show that if a (possibly non-binary and non-Boolean) predicate P is a singleton then P is not sparsifiable, our results show that the borderline of sparsifiability lies elsewhere. In particular, by Theorem 2, there are binary non-Boolean predicates that are not sparsifiable but are not singletons. Also, there are non-binary Boolean predicates that are not sparsifiable but are not singletons.

We remark that the term "sparsification" is also used in an unrelated line of work in which the goal is, given a CSP instance, to reduce the number of constraints without changing satisfiability of the instance; see, e.g., [6].

2 Classification of Binary Predicates

Throughout the paper we denote by $n = |V|$ the number of variables of a given CSP instance.

The following classification of binary Boolean predicates is from [7].

▶ **Theorem 1** ([7, Theorem 3.7]). *Let $P : \{0,1\}^2 \to \{0,1\}$ be a binary Boolean predicate. Let $0 < \varepsilon < 1$.*

1. *If P is a singleton then there exists an instance I of $\mathrm{CSP}(P)$ such that every ε-sparsifier of I has $\Omega(n^2)$ constraints.*
2. *Otherwise, for every instance I of $\mathrm{CSP}(P)$ there exists an ε-sparsifier of I with $O(\varepsilon^{-2}n)$ constraints.*

We denote by $\binom{D}{2} = \{B \subseteq D : |B| = 2\}$ the set of two-element subsets of D. For a binary predicate $P : D^2 \to \{0,1\}$ and $B, C \in \binom{D}{2}$, $P|_{B\times C}$ denotes the restriction of P onto $B \times C$.

The following is our main result, generalising Theorem 1 to arbitrary finite domains.

▶ **Theorem 2** (Main). *Let $P : D^2 \to \{0,1\}$ be a binary predicate, where D is a finite set with $|D| \geq 2$. Let $0 < \varepsilon < 1$.*

1. *If there exist $B, C \in \binom{D}{2}$ such that $P|_{B\times C}$ is a singleton then there exists an instance I of $\mathrm{CSP}(P)$ such that every ε-sparsifier of I has $\Omega(n^2)$ constraints.*
2. *Otherwise, for every instance I of $\mathrm{CSP}(P)$ there exists an ε-sparsifier of I with $O(\varepsilon^{-2}n)$ constraints.*

The rest of this section is devoted to proving Theorem 2.

First we introduce some useful notation. We set $[r] = \{0, 1, \dots, r-1\}$. We denote by $X \sqcup Y$ the disjoint union of X and Y. For any $r \geq 2$, we define $r\text{-}\mathsf{Cut} : [r]^2 \to \{0,1\}$ by $r\text{-}\mathsf{Cut}(x,y) = 1$ if and only if $x \neq y$.

Given an instance $I = (V, D, \Pi, w) \in \mathrm{CSP}(P)$, we denote by G^I the *corresponding graph* of I; that is, $G^I = (V, E, w)$ is a weighted directed graph with $E = \{(u,v) : ((u,v), P) \in \Pi\}$ and $w(u,v) = w((u,v), P)$. Conversely, given a weighted directed graph $G = (V, E, w)$ and a predicate $P : D^2 \to \{0,1\}$, the *corresponding* $\mathrm{CSP}(P)$ *instance* is $I^{G,P} = (V, D, \Pi, w)$, where $\Pi = \{(e, P) : e \in E\}$ and $w(e, P) = w(e)$. Hence, we can equivalently talk about instances of $\mathrm{CSP}(P)$ or (weighted directed) graphs. Thus, an ε-P-*sparsifier* of a graph $G = (V, E, w)$ is a subgraph $G_\varepsilon = (V, E_\varepsilon \subseteq E, w_\varepsilon)$ whose corresponding $\mathrm{CSP}(P)$ instance $I^{G_\varepsilon, P}$ is an ε-sparsifier of the corresponding $\mathrm{CSP}(P)$ instance $I^{G,P}$ of G.

Case (1) of Theorem 2 is established by the following result.

▶ **Theorem 3.** *Let $P : D^2 \to \{0,1\}$ be a binary predicate. Assume that there exist $B, C \in \binom{D}{2}$ such that $P|_{B\times C}$ is a singleton. For any n there is a* $\mathrm{CSP}(P)$ *instance I with $2n$ variables and n^2 constraints such that for any $0 < \varepsilon < 1$ it holds that any ε-sparsifier of I has n^2 constraints.*

Proof. Suppose $B = \{b, b'\}$, $C = \{c, c'\}$ and assume without loss of generality that ${P|_{B\times C}}^{-1}(1) = \{(b, c)\}$; that is, the support of $P|_{B\times C}$ is equal to $\{(b, c)\}$. Consider a $\mathrm{CSP}(P)$ instance $I = (V, D, \Pi, w)$, where

- $V = X \sqcup Y$, $X = \{x_1, \ldots, x_n\}$, and $Y = \{y_1, \ldots, y_n\}$;
- $\Pi = \{\pi_{ij} = ((x_i, y_j), P) : 1 \le i, j \le n\}$;
- w are arbitrary positive weights.

We have $|\Pi| = n^2$. We note that B and C may not be disjoint. We consider the family of assignments $A_{ij} : V \to B \cup C$ for $1 \le i, j \le n$ such that $A_{ij}(x_i) = b$, $A_{ij}(x) = b'$ for every $x \in X \setminus \{x_i\}$, $A_{ij}(y_j) = c$, and $A_{ij}(y) = c'$ for every $y \in Y \setminus \{y_j\}$. Then, we have

$$P(A_{ij}(u, v)) = \begin{cases} P(b, c) \;\; = 1 & \text{if } u = x_i, v = y_j, \\ P(b, c') \; = 0 & \text{if } u = x_i, v \in Y \setminus \{y_j\}, \\ P(b', c) \; = 0 & \text{if } u \in X \setminus \{x_i\}, v = y_j, \\ P(b', c') = 0 & \text{if } u \in X \setminus \{x_i\}, v \in Y \setminus \{y_j\}. \end{cases}$$

Therefore,

$$\mathrm{Val}_I(A_{ij}) = \sum_{\pi \in \Pi} w(\pi) P(A_{ij}(\pi)) = w(\pi_{ij}) > 0.$$

Hence, if $I_\varepsilon = (V, D, \Pi_\varepsilon, w_\varepsilon)$ is an ε-sparsifier of I, we must have that $\pi_{ij} \in \Pi_\varepsilon$ for every $1 \le i, j \le n$, as otherwise we would have

$$\mathrm{Val}_{I_\varepsilon}(A_{ij}) = \sum_{\pi \in \Pi_\varepsilon} w_\varepsilon(\pi) P(A_{ij}(\pi)) = 0 \notin (1 \pm \varepsilon) \mathrm{Val}_I(A_{ij}).$$

Therefore, we have $\Pi_\varepsilon = \Pi$ and hence $|\Pi_\varepsilon| = |\Pi| = n^2$. ◀

The main tool used in the proof of Theorem 1 (2) from [7] is a graph transformation known as the bipartite double cover [5], which allows for a reduction to cut sparsifiers [3].

▶ **Definition 4.** *For a weighted directed graph $G = (V, E, w)$, the* bipartite double cover *of G is the weighted directed graph $\gamma(G) = (V^\gamma, E^\gamma, w^\gamma)$, where*

- $V^\gamma = \{v^{(0)}, v^{(1)} : v \in V\}$;
- $E^\gamma = \{(u^{(0)}, v^{(1)}) : (u, v) \in E\}$;
- $w^\gamma(u^{(0)}, v^{(1)}) = w(u, v)$.

Given an assignment $A : V \to [r]$, we let $\mathcal{A} = (A_0, \ldots, A_{r-1})$ be the induced r-partition of V, where $A_j = A^{-1}(j)$. For a binary predicate $P : [r]^2 \to \{0,1\}$ and an instance $I = (V, [r], \Pi, w) \in \mathrm{CSP}(P)$, we define $\mathrm{Val}_I(\mathcal{A}) = \mathrm{Val}_I(A)$. Moreover, for a weighted directed graph G and a binary predicate P, we define $\mathrm{Val}_{G,P}(\mathcal{A}) = \mathrm{Val}_{I^{G,P}}(\mathcal{A})$. We denote the set of all r-partitions of V by $Part_r(V)$.

For any r-partition $\mathcal{A} = (A_0, \ldots, A_{r-1})$ of the vertices of V, let $A_i^{(j)} = \{v^{(j)} : v \in A_i\}$. Thus $\mathcal{A}^\gamma = (A_0^{(0)}, A_0^{(1)}, \ldots, A_{r-1}^{(0)}, A_{r-1}^{(1)})$ is a $2r$-partition of the vertices of V^γ.

We use an argument from the proof of Theorem 1 (2) from [7] and apply it to non-Boolean predicates.

▶ **Proposition 5.** *Let $P:[r]^2\to\{0,1\}$ and $P':[r']^2\to\{0,1\}$ be binary predicates. Suppose that there is a function $f_P: Part_r(V)\to Part_{r'}(V^\gamma)$ such that for any weighted directed graph G on V and for any r-partition $\mathcal{A}\in Part_r(V)$ it holds that*

$$\mathrm{Val}_{G,P}(\mathcal{A})=\mathrm{Val}_{\gamma(G),P'}(f_P(\mathcal{A})),$$

where $\gamma(G)=(V^\gamma,E^\gamma,w^\gamma)$ is the bipartite double cover of G. If there is an ε-P'-sparsifier of $\gamma(G)$ of size $g(n)$ then there is an ε-P-sparsifier of G of size $O(g(n))$.

Proof. Given $G=(V,E,w)$, let $\gamma(G)_\varepsilon=(V,E^\gamma_\varepsilon,w^\gamma_\varepsilon)$ be an ε-P'-sparsifier of the bipartite double cover $\gamma(G)$ of G. Define a subgraph $G_\varepsilon=(V,E_\varepsilon,w_\varepsilon)$ of G by $E_\varepsilon=\{(u,v):(u^{(0)},v^{(1)})\in E^\gamma_\varepsilon\}$ and $w_\varepsilon(u,v)=w^\gamma_\varepsilon(u^{(0)},v^{(1)})$. Notice that $\gamma(G_\varepsilon)=\gamma(G)_\varepsilon$, $E_\varepsilon\subseteq E$, and $E^\gamma=O(|E|)$.

Then, we have

$$\begin{aligned}\mathrm{Val}_{G_\varepsilon,P}(\mathcal{A})&=\mathrm{Val}_{\gamma(G_\varepsilon),P'}(f_P(\mathcal{A}))\\&=\mathrm{Val}_{\gamma(G)_\varepsilon,P'}(f_P(\mathcal{A}))\in(1\pm\varepsilon)\,\mathrm{Val}_{\gamma(G),P'}(f_P(\mathcal{A}))=(1\pm\varepsilon)\,\mathrm{Val}_{G,P}(\mathcal{A})\end{aligned}$$

and

$$|E_\varepsilon|\le|E^\gamma_\varepsilon|=O\Big(\frac{|V^\gamma|}{\varepsilon^2}\Big)=O\Big(\frac{|V|}{\varepsilon^2}\Big),$$

implying that G_ε is also an ε-P-sparsifier of G.

Moreover, $|E_\varepsilon|\le|E^\gamma_\varepsilon|=g(n)$ implies $|E_\varepsilon|=O(g(n))$. ◀

We now focus on proving Case (2) of Theorem 2. Assume that for any $B,C\in\binom{D}{2}$, $P|_{B\times C}$ is not a singleton. Our strategy is to show that in this case the value of a CSP(P) instance under any assignment can be expressed as the value of a corresponding CSP(ℓ-Cut) instance (for some $\ell\le 2|D|$) under the same assignment.

For an undirected graph $G=(V,E)$ and a subset $U\subseteq V$, we denote the vertex-induced subgraph on U by $G[U]$ and its edge set by $E[U]$. For a possibly disconnected undirected graph G, we denote the connected component containing a vertex v by $G_v=(V(G_v),E(G_v))$. Finally, we denote the degree of vertex v in graph G by $d_G(v)$.

▶ **Definition 6.** *Let $G=(U\sqcup V,E)$ be an undirected bipartite graph. The* bipartite complement *$\overline{G}=(U\sqcup V,\overline{E})$ of G has the following edge set:*

$$\overline{E}=\{\{u,v\}:u\in U,v\in V,\{u,v\}\notin E\}.$$

The following property of bipartite graphs will be crucial in the proof of Theorem 8.

▶ **Proposition 7.** *Let $G=(U\sqcup V,E)$ be a bipartite graph with $|U|=|V|=r$, $r\ge 2$. Assume that for any $u,u'\in U$ and $v,v'\in V$ we have $|E[\{u,u',v,v'\}]|\neq 1$. Then, for any $v\in U\sqcup V$ with $d_{\overline{G}}(v)>0$, $\overline{G}_v$ is a complete bipartite graph with partition classes $\{U\cap V(\overline{G}_v)\}$ and $\{V\cap V(\overline{G}_v)\}$.*

Proof. For contradiction, assume that there are $u\in U$ and $v\in V$ such that $\{u,v\}\notin\overline{E}$ but u and v belong to the same connected component of $\overline{G}$. Choose u and v with the shortest possible distance between them. Let $u=u_0,u_1,\dots,u_k=v$ be a shortest path between u and v in $\overline{G}$, where $k\ge 3$ is odd. We will show that $|\overline{E}[\{u_0,u_1,u_{k-1},u_k\}]|=3$, which contradicts the assumption that $|E[\{u_0,u_1,u_{k-1},u_k\}]|\neq 1$.

If $k=3$ then the claim holds since we assumed that $\{u_0,u_1\},\{u_1,u_2\},\{u_2,u_3\}\in\overline{E}$ and $\{u_0,u_3\}\notin\overline{E}$.

Let $k \geq 5$. We will be done if we show that $\{u_1, u_{k-1}\} \in \overline{E}$, as by our assumptions $\{u_0, u_1\}, \{u_{k-1}, u_k\} \in \overline{E}$ and $\{u_0, u_k\} \notin \overline{E}$. To this end, note that $\{u_0, u_{k-2}\} \in \overline{E}$ as otherwise u_0 and u_{k-2} would be a pair of vertices with the required properties but of distance $k-2$, contradicting our choice of u and v. Thus, $\{u_1, u_{k-1}\} \in \overline{E}$ as otherwise we would have $|\overline{E}[\{u_0, u_1, u_{k-2}, u_{k-1}\}]| = 3$, which contradicts $|E[\{u_0, u_1, u_{k-2}, u_{k-1}\}]| \neq 1$. ◀

Case (2) of Theorem 2 is established by the following result.

▶ **Theorem 8.** *Let $P : D^2 \to \{0,1\}$ be a binary predicate such that for any $B, C \in \binom{D}{2}$ we have that $P|_{B\times C}$ is not a singleton. Then, for every $0 < \varepsilon < 1$ and every instance I of* CSP(P) *there is a sparsifier of I with $O(\varepsilon^{-2}n)$ constraints.*

Proof. Let $I = (V, D, \Pi, w)$ be an instance of CSP(P) with $r = |D|$. Without loss of generality, we assume that $D = [r]$. Let $G = G^I = (V, E, w)$ be the corresponding (weighted directed) graph of I, and let $\gamma(G) = (V^\gamma, E^\gamma, w^\gamma)$ be the bipartite double cover of G. Recall that for an assignment $A : V \to [r]$, we denote $A_i = A^{-1}(i)$. Thus, $\mathcal{A} = (A_0, \ldots, A_{r-1})$ forms an r-partition of V.

Our goal is to show the existence of a function $f_P : Part_r(V) \to Part_\ell(V^\gamma)$ (for some fixed $\ell \leq 2r$) such that

$$\forall A : V \to [r], \qquad \mathrm{Val}_{G,P}(\mathcal{A}) = \mathrm{Val}_{\gamma(G),\ell\text{-}\mathsf{Cut}}(f_P(\mathcal{A})). \tag{1}$$

Assuming the existence of f_P, we can finish the proof as follows. Batson, Spielman, and Srivastava established the existence of a sparsifier of size $O(\varepsilon^{-2}n)$ for any instance of CSP(2-Cut) [3]. By [7, Section 6.2], this implies the existence of a sparsifier of size $O(\varepsilon^{-2}n)$ for any instance of CSP(ℓ-Cut). Consequently, by Proposition 5 and (1), there is a sparsifier of size $O(\varepsilon^{-2}n)$ for the instance $I^{G,P} = I$.

It remains to show the existence of f_P satisfying (1).

In the proof of Theorem 1 (2) in [7], such functions are given for a binary Boolean predicate P with support size $|P^{-1}(1)| \in \{0, 2, 4\}$. In what follows we give a construction of f_P for an *arbitrary* binary predicate $P : [r]^2 \to \{0,1\}$ with $r \geq 2$ from the statement of the theorem.

Although the bipartite double cover is commonly defined as a directed graph, in this proof we will consider the *undirected* bipartite double cover $\gamma(G)$ of G.[5] We also define an auxiliary graph $G^P = (V^P, E^P)$, where

$$V^P = \{v_0, v_0', \ldots, v_{r-1}, v_{r-1}'\},$$
$$E^P = \{\{v_i, v_j'\} : P(i,j) = 1\}.$$

Let ℓ be the number of connected components of $\overline{G^P}$, the bipartite complement of G^P. By definition, $\ell \leq |V^P| = 2r$.

We need to find a function $f_P : Part_r(V) \to Part_\ell(V^\gamma)$ that satisfies (1) for all $\mathcal{A} \in Part_r(V)$. Such a function corresponds to a map $c : V^P \to [\ell]$ on the vertices of G^P with the following property:

$$\forall i, j \in [r] \quad \begin{cases} \{v_i, v_j'\} \in E^P \implies c(v_i) \neq c(v_j') \\ \{v_i, v_j'\} \notin E^P \implies c(v_i) = c(v_j'). \end{cases}$$

[5] We had defined the bipartite double cover as a directed graph. However, here it is easier to deal with undirected graphs, as since ℓ-Cut is a symmetric predicate, the direction of the edges makes no difference. Furthermore, notice that by the way the bipartite double cover is constructed, removing the direction does *not* turn the graph into a multigraph.

We call such maps *colourings*. Indeed, the colouring c induces, for $\mathcal{A}$, an assignment $A^\gamma : V^\gamma \to [\ell]$ of the vertices of $\gamma(G)$ which satisfies

$$A^\gamma(u) = c(v_{A(u)}) \quad \text{and} \quad A^\gamma(u') = c(v'_{A(u)})$$

and which, in turn, induces a partition $\{U_i\}_{i=0}^{\ell-1}$ of V^γ with $U_i = (A^\gamma)^{-1}(i)$. We define $f_P(\mathcal{A}) = (U_0, \ldots, U_{\ell-1})$. Now for any $u, v \in V$ and for any assignment $A : V \to [r]$, we have

$$\begin{aligned} P(A(u), A(v)) = 1 &\iff \{v_{A(u)}, v'_{A(v)}\} \in E^P \\ &\iff c(v_{A(u)}) \neq c(v'_{A(v)}) \\ &\iff A^\gamma(u) \neq A^\gamma(v') \\ &\iff \ell\text{-}\mathsf{Cut}(A^\gamma(u), A^\gamma(v')) = 1. \end{aligned}$$

Moreover, by the definition of the graph bipartite double cover, we have $w(u, v) = w^\gamma(u, v')$ for all $u, v \in V$, implying that

$$\begin{aligned} \mathrm{Val}_{G,P}(\mathcal{A}) = \mathrm{Val}_{G,P}(A_0, \ldots, A_{r-1}) &= \sum_{(u,v)\in E} w(u, v)P(A(u), A(v)) \\ &= \sum_{(u,v')\in E^\gamma} w^\gamma(u, v')\ell\text{-}\mathsf{Cut}(A^\gamma(u), A^\gamma(v')) = \mathrm{Val}_{\gamma(G),\ell\text{-}\mathsf{Cut}}(A^\gamma) \\ &= \mathrm{Val}_{\gamma(G),\ell\text{-}\mathsf{Cut}}(U_0, \ldots, U_{\ell-1}) = \mathrm{Val}_{\gamma(G),\ell\text{-}\mathsf{Cut}}(f_P(\mathcal{A})) \end{aligned}$$

as required.

While a colouring does not exist for an arbitrary bipartite graph, we now argue that a colouring does exist if the auxiliary graph G^P arises from a predicate P from the statement of the theorem. Since for any $B, C \in \binom{[r]}{2}$ we have $|P|_{B\times C}{}^{-1}(1)| \neq 1$, G^P satisfies the assumptions of Proposition 7. Therefore, the ℓ separate connected components which form its bipartite complement $\overline{G^P}$ are complete bipartite graphs. We can assign one of the ℓ colours to each connected component to get a colouring for the graph G^P. ◀

3 Conclusion

For simplicity, we have only presented our main result on binary CSPs over a single domain. However, it is not difficult to extend our result to the so-called *multisorted* binary CSPs, in which different variables come with possibly different domains.

We have classified *binary* CSPs (on finite domains) but much more work seems required for a full classification of non-binary CSPs. We have made some initial steps.

For any $k \geq 3$, the k-ary Boolean "not-all-equal" predicate $k\text{-}\mathsf{NAE} : \{0, 1\}^k \to \{0, 1\}$ is defined by $k\text{-}\mathsf{NAE}^{-1}(0) = \{(0, \ldots, 0), (1, \ldots, 1)\}$. Kogan and Krauthgamer showed that the k-NAE predicates, which correspond to hypergraph cuts, are sparsifiable [9, Theorem 3.1]. By extending bipartite double covers for graphs in a natural way to k-partite k-fold covers, we obtain sparsifiability for the class of k-ary predicates that can be rewritten in terms of k-NAE. On the other hand, we identify a whole class of predicates that are not sparsifiable, namely those k-ary predicates that contain a singleton ℓ-cube for some $\ell \leq k$. However, there are predicates which do not fall in either of these two categories; that is, predicates that cannot be proved sparsifiable via k-partite k-fold covers but also cannot be proved non-sparsifiable via the current techniques. An example of such predicates are the "parity" predicates.

References

1 Kook Jin Ahn and Sudipto Guha. Graph Sparsification in the Semi-streaming Model. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09), Part II*, volume 5556 of *Lecture Notes in Computer Science*, pages 328–338. Springer, 2009. doi:10.1007/978-3-642-02930-1_27.

2 Alexandr Andoni, Jiecao Chen, Robert Krauthgamer, Bo Qin, David P. Woodruff, and Qin Zhang. On Sketching Quadratic Forms. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science (ITCS'16)*, pages 311–319, 2016. doi:10.1145/2840728.2840753.

3 Joshua D. Batson, Daniel A. Spielman, and Nikhil Srivastava. Twice-Ramanujan Sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012. doi:10.1137/090772873.

4 András A. Benczúr and David R. Karger. Approximating s-t Minimum Cuts in $\tilde{O}(n^2)$ Time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing (STOC'96)*, pages 47–55, 1996. doi:10.1145/237814.237827.

5 Richard A. Brualdi, Frank Harary, and Zevi Miller. Bigraphs versus digraphs via matrices. *Journal of Graph Theory*, 4(1):51–73, 1980. doi:10.1002/jgt.3190040107.

6 Hubie Chen, Bart M. P. Jansen, and Astrid Pieterse. Best-case and Worst-case Sparsifiability of Boolean CSPs. In *Proceedings of the 13th International Symposium on Parameterized and Exact Computation (IPEC'18)*, 2018. arXiv:1809.06171.

7 Arnold Filtser and Robert Krauthgamer. Sparsification of Two-Variable Valued Constraint Satisfaction Problems. *SIAM Journal on Discrete Mathematics*, 31(2):1263–1276, 2017. doi:10.1137/15M1046186.

8 Wai Shing Fung, Ramesh Hariharan, Nicholas J. A. Harvey, and Debmalya Panigrahi. A general framework for graph sparsification. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC'11)*, pages 71–80. ACM, 2011. doi:10.1145/1993636.1993647.

9 Dmitry Kogan and Robert Krauthgamer. Sketching Cuts in Graphs and Hypergraphs. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science (ITCS'15)*, pages 367–376, 2015. doi:10.1145/2688073.2688093.

10 Vladimir Kolmogorov, Andrei A. Krokhin, and Michal Rolínek. The Complexity of General-Valued CSPs. *SIAM Journal on Computing*, 46(3):1087–1110, 2017. doi:10.1137/16M1091836.

11 Ilan Newman and Yuri Rabinovich. On Multiplicative Lambda-Approximations and Some Geometric Applications. *SIAM Journal on Computing*, 42(3):855–883, 2013. doi:10.1137/100801809.

12 Tasuko Soma and Yuichi Yoshida. Spectral Sparsification of Hypergraphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*, 2019.

13 Daniel A. Spielman and Nikhil Srivastava. Graph Sparsification by Effective Resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. doi:10.1137/080734029.

14 Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC'04)*, pages 81–90. ACM, 2004. doi:10.1145/1007352.1007372.

15 Daniel A. Spielman and Shang-Hua Teng. Spectral Sparsification of Graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011. doi:10.1137/08074489X.